POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Systems and Methods for Big and Unstructured Data Project

Author(s): **Gabriele Ginestroni**
Giacomo Gumiero
Lorenzo Iovine
Nicola Landini
Francesco Leone

Group Number: **10**

Academic Year: 2022-2023

Contents

1 | Introduction

In this chapter will be presented the problem specification and the hypothesis under which the database is implemented.

1.1. Problem Specification

This project aims to build an Information System that handles scientific articles contained in the DBLP bibliography. The project involves managing the type of the articles and the associated DOI (Digital Object Identifier), which identifies a publication or a document and links to it on the web. Other entities to deal with are authors, identified by an ID or ORCID (Open Researcher and Contributor ID), and their affiliations with organizations. In order to address the problem, we will store data in a graph database, allowing us to visualize relations and handle information correctly.

1.2. Assumptions

1. All the data in the dataset are heterogeneous, so fields are different
2. The **authors** with missing field *_id* are not considered
3. It is possible that an author writes for different organizations
4. Field *_id* in **author** is unique
5. Field *_id* in **article** exists and it is unique
6. It is impossible that 2 different articles are on the same journal, in the same *volume* with an intersection between *page_start* and *page_end*
7. The designed model doesn't take into consideration the URL associated to the article node, as the main focus of the project was not reading the article
8. It is possible to find a self-reference in a publication
9. A *venue* can be instantiated as a journal, a conference or a generic venue!!!!!!!!

2 | ER Diagram

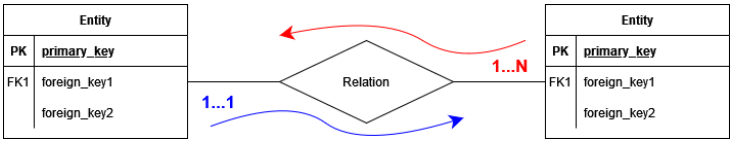

Figure 2.1: ER Diagram Organization

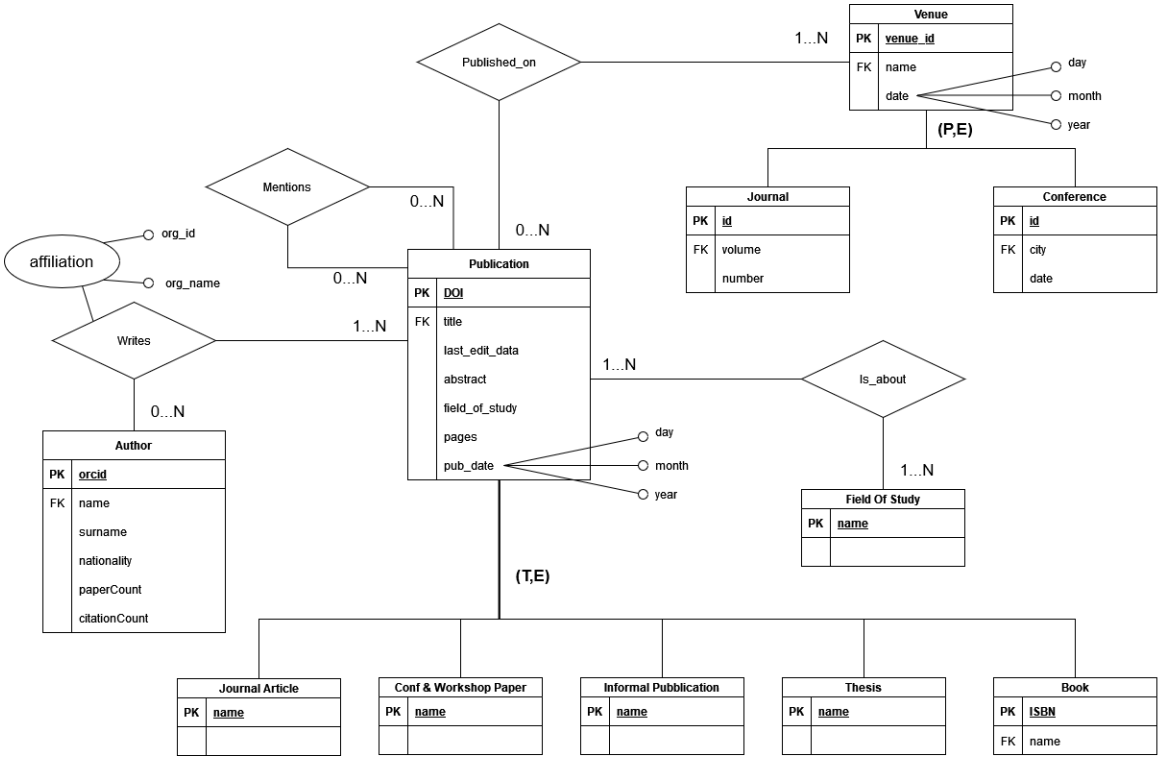

Figure 2.2: ER Diagram

The ER diagram designed contains the following entities:

- **Publication:** this entity represents all the scientific articles. They are identified by their primary key *_id* and other important attributes are: *DOI, title, last_edit_data, abstract, keywords, pages, pub_date*. Of course the attributes of such entity could be enlarged, but as a sample dataset we have believed these are enough. Publication entity is the superclass of a Total and Exclusive ISA relationship with the following subclasses: *Journal article, Conference & Workshop paper, Informal publication, Thesis, Book*
- **Author:** it represents all the people that submitted at least one publication. Its primary key is *_id* and the foreign keys are: *name, surname, nationality, paperCount, citationCount*. Of course the attributes of such entity could be enlarged, but as a sample dataset we have believed these are enough
- **Venue:** it's the entity that represents the type of a publication. This is a superclass that creates a Partial and Exclusive ISA relationship with the two subclasses *Journal* and *Conference*. The primary key is *raw* and the other keys are: *name, date, venue_id*
- **Field Of Study:** this entity represents the topics of the related publication

The ER diagram designed contains the following relationships:

- **Writes:** is the relationship between *author* and *publication* which specifies also the affiliation with the org_id and the org_name
- **Mentions:** occurs between two *publication* and specifies when a publication refers to another one
- **Is about:** binds a *publication* with its *fields of study*
- **Published on:** simply relates a *publication* to its *venue*

3 | Dataset Description

3.1. Dataset Preprocessing

The dataset we used is based on DBLP-Citation-network V13 at https://www.aminer.org/citation whose size is 13GB. Using **Pandas Profiling** we analyzed the dataset focusing our attention on distinctness and completeness of the attributes. We used this result for selecting primary keys (attributes with high values of distinctness and low missing values), to ignore fields that were not informative, and also to filter the tuples. Then we sliced the entire dataset obtaining a subset of 16MB. The slicing operation was not performed randomly in order not to obtain a subset of the dataset containing some publications but any of the publications referenced by them. The working sample of 6400 publications was obtained in this way:

- first a partition of 20M rows was extracted from the original database so that we started from a pool of 257K articles instead of using all the original database of 5.4M articles, we used 5% as a sample to speed up the sample generation process
- then an extract of 4K publications was used as a base, we arbitrarily picked the ones from position 20000 to 24000
- in the end the ids of those 4K publications were looked up inside our partition to obtain also articles that cited them, in order to have sufficient relationships to make different queries

The following is a part of the script used for the operation just described:

```
sed -n $start","$end"p" $input > tmp.json
while IFS='' read -r line || [[ -n "$line" ]]; do
  artId=$(echo "$line" | cut -d '"' -f4)
  grep "$artId" $input >> $output
done < tmp.json
```

In the end, we have to say that we didn't add extra data to our slice of the dataset in order to maintain the coherency of the information.

3.2. Attributes Description

In this section we will present all the attributes contained, pointing which of them are considered or not.

3.2.1. Publication

Publication represent the central concept of the system and contains:

- **_id** is an alphanumeric string that is the primary key, that's because is unique and every nodes owns this parameter
- **title** represents the title of the publication
- **authors** is an array of authors that will be presented later
- **venue** defines an entity that will be presented later
- **year** represents the year of publication
- **keywords** is an array containing the tag of subjects faced in the publication
- **fos** is an array containing the fields of study of the publication
- **n_citation** is the number of times that this publication has been mentioned
- **page_start** defines the starting page of the publication. This attribute wasn't take into consideration because doesn't target the goal of the project
- **page_end** defines the last page of the publication. This attribute wasn't take into consideration because doesn't target the goal of the project
- **lang** represents the language of the publication
- **volume** is the volume of the publication. This attribute is used in the relationship between *Article* and *Venue*
- **issue** refers to how many times a periodical has been published during that year. This attribute wasn't take into consideration due to the presence of many missing or null values
- **issn** is an identification code of the publication. This attribute wasn't take into consideration due to the presence of many missing or null values
- **isbn** is an identification code of the publication. This attribute wasn't take into consideration due to the presence of many missing or null values

- **doi** Digital Object Identifier is a persistent identifier. We decided to take it into consideration due to an acceptable missing percentage, much lower than the one affecting issn or isbn attributes
- **pdf** contains a string that links to the publication PDF online. This attribute wasn't take into consideration due to the presence of many missing or null values
- **url** contains an array of links to the publication resources online. This attribute wasn't take into consideration because doesn't target the goal of the project
- **abstract** is a string containing a brief summary of the contents of the paper
- **references** is an array of ids representing the publication mentioned

3.2.2. Author

Author is the most present entity of the system and contains:

- **_id** is an alphanumeric string that is the primary key, that's because is unique and almost every nodes owns this parameter
- **name** is the name of the author
- **org** is a string that represents the organization in which the author works. It is used as an attribute of the relationship *Writes* described before
- **gid** is an identifier that represents the organization in which the author works. This attribute wasn't take into consideration due to the presence of many missing or null values
- **orgid** is an identifier that represents the organization in which the author works. It is used as an attribute of the relationship *Writes* described before
- **orgs** is an array of organizations for which the author worked. This attribute wasn't take into consideration due to the presence of many missing or null values
- **email** is a string containing the email address of the author. This attribute wasn't take into consideration due to the presence of many missing or null values and because doesn't target the goal of the project
- **orcid** Open Researcher and Contributor ID is a unique identifier for authors of scientific articles. This attribute is taken into consideration although is not always present

- **oid** is an identifier for the author. This attribute wasn't take into consideration due to the presence of many missing or null values
- **bio** is a string that describes the author. This attribute wasn't take into consideration due to the presence of many missing or null values
- **sid** is an identifier for the author. This attribute wasn't take into consideration due to the presence of many missing or null values
- **name_zh** is the name of the organization in which the author works. This attribute wasn't take into consideration due to the presence of many missing or null values
- **org_zh** is an identifier that represents the organization in which the author works. This attribute wasn't take into consideration due to the presence of many missing or null values

3.2.3. Venue

Venue is the entity that represents the type of a publication. Thanks to the profiling, we found that venue nodes contains the following attributes:

- **_id** is an alphanumeric identifier. This attribute is not used as a primary key due to the large amount of missing values
- **raw** is the name or the abbreviation of the event or volume (without specifying the year) in which the publication was presented. This attribute was chosen as the primary key thanks to the low number of missing values and to the fact that in this way publications were grouped based on the event in which they were presented
- **raw_zh** refers to the event (without specifying the year) in which the publication was presented. This attribute wasn't take into consideration due to the presence of many missing or null values
- **type** indicates the type of the publication. Exploiting the profiling we understood the distribution of different values of *type*, and become clear that 0 and 1 were attributable respectively to conference and journal. The other values were not easily attributable to other types of publication, so we decided to consider as a generic venue:
  - every *type* values different from 0 or 1
  - every entity in which the *type* field is missing

- **sid** name of the journal in which the publication was published. This attribute wasn't take into consideration due to the presence of many missing or null values
- **t** represents the type of the publication. This attribute wasn't take into consideration due to the presence of many missing or null values
- **issn** is an identification code of the publication
- **name_d** is the extended name of the event or volume (without specifying the year) in which the publication was presented
- **publisher** is a string containing the name of the publisher
- **online_issn** is an identification code of the publication

4 | Graph Diagram

5 | Sample Dataset

In this chapter we will present the import commands. In order to complete them we used the plug-in apoc.

1. Create *Publication*, *Author* and `WRITES` relationship between them:

```
call apoc.load.json("test.json") yield value
UNWIND value as pub
UNWIND pub.authors as aut WITH aut,pub where aut._id is not null
MERGE (publication:Publication{id:pub._id}) ON CREATE SET
    publication.title  = pub.title,
    publication.doi = pub.doi,
    publication.year = pub.year,
    publication.n_citation = pub.n_citation,
    publication.keywords = pub.keywords,
    publication.abstract = pub.abstract,
    publication.lang = pub.lang
MERGE(author:Author{id:aut._id}) ON CREATE SET
    author.name = aut.name,
    author.orcid = aut.orcid
MERGE(author)-[writes:WRITES]->(publication) ON CREATE SET
    writes.org=aut.org,
    writes.orgid=aut.orgid

Added 23365 labels, created 23365 nodes, set 138863 properties,
   created 18534 relationships, completed after 104197 ms.
```

2. Create `REFERENCE` relationship between *Articles*:

```
call apoc.load.json("test.json") yield value
UNWIND value as pub
UNWIND pub.references as ref
MATCH (init:Publication{id:pub._id})
MATCH (final:Publication{id:ref})
MERGE(init)-[:REFERENCES]->(final)

Created 2866 relationships, completed after 466136 ms.
```

3. Create *Conference* when `raw` exists and `type` equal to 0:

```
call apoc.load.json("test.json") yield value
UNWIND value as art
WITH art where art.venue.raw is not null and art.venue.type = 0
MATCH(article:Publication{id: art._id})
MERGE(venue:Conference{raw:art.venue.raw}) ON CREATE SET
    venue.name = art.venue.name_d
MERGE(article)-[pub:PUBLISHED]->(venue) ON CREATE SET
  pub.issue = art.venue.issue,
    pub.volume = art.venue.volume,
  pub.issn = art.venue.issn,
  pub.online_issn = art.venue.online_issn,
  pub.publisher = art.venue.publisher

Added 2133 labels, created 2133 nodes, set 22676 properties,
   created 4846 relationships, completed after 22029 ms.
```

4. Create *Journal* when `raw` exists and `type` equal to 1:

```
call apoc.load.json("test.json") yield value
UNWIND value as art
WITH art where art.venue.raw is not null and art.venue.type = 1
MATCH(article:Publication{id:art._id})
MERGE(venue:Journal{raw:art.venue.raw}) ON CREATE SET
    venue.name = art.venue.name_d
MERGE(article)-[pub:PUBLISHED]->(venue) ON CREATE SET
  pub.issue = art.venue.issue,
    pub.volume = art.venue.volume,
  pub.issn = art.venue.issn,
  pub.online_issn = art.venue.online_issn,
  pub.publisher = art.venue.publisher

Added 256 labels, created 256 nodes, set 1681 properties, created
   351 relationships, completed after 6576 ms.
```

5. Create *Generic Venue* when `raw` doesn't exist or `type` different from 0 or 1:

```
call apoc.load.json("test.json") yield value
UNWIND value as art
WITH art where art.venue.raw is not null and (art.venue.type is
   null or (art.venue.type <> 1 and art.venue.type <> 0))
MATCH(article:Publication{id:art._id})
MERGE(venue:GenericVenue{raw:art.venue.raw}) ON CREATE SET
    venue.name = art.venue.name_d
MERGE(article)-[pub:PUBLISHED]->(venue) ON CREATE SET
```

```
  pub.issue = art.venue.issue,
    pub.volume = art.venue.volume,
  pub.issn = art.venue.issn,
  pub.online_issn = art.venue.online_issn,
  pub.publisher = art.venue.publisher

Added 586 labels, created 586 nodes, set 5122 properties, created
   1116 relationships, completed after 8705 ms.
```

6. Create *Field of Study* and `REGARD` relationship between *Field of Study* and *Publication*:

```
call apoc.load.json("test.json") yield value
UNWIND value as pub
UNWIND pub.authors as aut WITH aut,pub where aut._id is not null
MERGE (publication:Publication{id:pub._id}) ON CREATE SET
    publication.title  = pub.title,
    publication.doi = pub.doi,
    publication.year = pub.year,
    publication.n_citation = pub.n_citation,
    publication.keywords = pub.keywords,
    publication.abstract = pub.abstract,
    publication.lang = pub.lang
MERGE(author:Author{id:aut._id}) ON CREATE SET
    author.name = aut.name,
    author.orcid = aut.orcid
MERGE(author)-[writes:WRITES]->(publication) ON CREATE SET
    writes.org=aut.org,
    writes.orgid=aut.orgid

Added 23365 labels, created 23365 nodes, set 138863 properties,
   created 18534 relationships, completed after 104197 ms.
```

6 | Queries and Commands

7 | Conclusion

A | Appendix A

If you need to include an appendix to support the research in your thesis, you can place it at the end of the manuscript. An appendix contains supplementary material (figures, tables, data, codes, mathematical proofs, surveys, ...) which supplement the main results contained in the previous chapters.

List of Figures

List of Tables